



Anekant Education Society's

**Tuljaram Chaturchand College
of Arts, Science & Commerce, Baramati**

(AUTONOMOUS)

Affiliated

Savitribai Phule Pune University, Pune

Department Of Computer Science

F.Y.B.Sc.(Computer Science)

Lab Course I (CSCO-1203) – Advanced 'C'

Lab Course II (CSCO -1204) – DBMS - II

(Semester I)

20 - 20

Name _____

Roll No. _____ Division _____

Anekant Education Society's
Tuljaram Chaturchand College of
Arts, Science & Commerce Baramati (Autonomous)
DEPARTMENT OF COMPUTER SCIENCE

Computer Practical Journal
Paper III: CSC – 1203 Lab Course on Advanced C

- CERTIFICATE -

University Seat No.

Date :

This is to certify that Mr./ Miss. / Smt. _____
_____ has satisfactorily completed the course in practical as
prescribed by the University of Pune for the **F. Y. B.Sc.(Computer Science-Lab
Course I** in the Year 2019 – 2020

In-Charge
(Practical)

Internal Examiner

External Examiner

Head
Computer Science

Index

Sr. No.	Assignment Name	Date	Remark
1	Assignment to demonstrate use of pointers		
2	Assignment to demonstrate concept of strings (string & pointers)		
3	Assignment to demonstrate array of strings		
4	Assignment to demonstrate use of bitwise operators		
5	Assignment to demonstrate structures (using array and functions)		
6	Assignment to demonstrate structures and unions		
7	Assignment to demonstrate command line arguments and preprocessor directives		
8	Assignment to demonstrate file handling (text files)		
9	Assignment to demonstrate file handling (binary files and random access to files)		
10	Assignment to demonstrate graphics programming		
	C Programming – Case Study		

Assignment:1

Assignment to demonstrate use of pointers

A Pointer is a variable that stores the memory address of another variable

Actions involving Pointers	Syntax	Example
Declaration of pointers	<code>data_type * pointer_name</code>	<code>int *p1,*p2; float *temp1;</code>
Initialization of pointers	<code>pointer =&variable p1=&n;</code>	<code>int a, *p= &a;</code>
Pointer Arithmetic	The C language allow arithmetic operations to be performed on pointers: Increment, Decrement, Addition, Subtraction When a pointer is incremented (or decremented) by 1, it increments by <code>sizeof(datatype)</code> . For example, an integer pointer increments by <code>sizeof(int)</code> .	
Pointers and Functions	We can pass the address of a variable to a function. The function can accept this address in a pointer and use the pointer to access the variable's value.	
Arrays And Pointers	An array name is a pointer to the first element in the array. It holds the base address of the array. Every array expression is converted to pointer expression as follows: <code>a[i]</code> is same as <code>*(a+i)</code>	<code>int n;</code> <code>*n , *(n + 0)</code> represents 0 th element <code>n[j], *(n+ j),* (j + n) , j [n] :</code> represent the value of the j th element of array n
Pointer To Pointer		<code>int a; int * p; int **q; p = &a; q = *p ;</code>

<p>To allocate memory Dynamically</p>	<p>The functions used are : malloc, calloc, realloc ptr = (cast-type *) malloc (byte-size) ; Allocates a block of contiguous bytes. If the space in heap is not sufficient to satisfy request, allocation fails, returns NULL. ptr1 = (cast-type *) calloc (byte-size); Similar to malloc, but initializes the memory block allocated to 0. ptr = realloc (ptr, new size); To increase / decrease memory size.</p>	<pre>int * p,*p1; p = (int *) malloc(10 * sizeof(int)); p1 = (int *) calloc(10, sizeof(int)); p1=realloc(p1,20*sizeof(int));</pre>
---------------------------------------	---	--

Write Programs to solve the following problems

Set – A

1. Write a C program to Accept data & display its addresses.
2. Write a C program to Accept array elements & perform sum of array elements using pointers.
3. Write a program to allocate memory dynamically for n integer Accept & displays the values.

Set – B

1. Write a function which takes hours, minutes and seconds as parameters and an integer s and increments the time by s seconds. Accept time and seconds in main and Display the new time in main using the above function.
2. Write a program to display the elements of an array containing n integers in the reverse order using a pointer to the array.

Signature of the Instructor

Remark

Date

Assignment:2

Assignment to demonstrate concept of strings (string & pointers)

A string is an array of characters terminated by a special character called NULL character(\0).

Each character is stored in 1 byte as its ASCII code.

Actions Involving strings	Explanation	Example
Declaring Strings		char message[80];
Initializing Strings		char message[] = { 'H', 'e', 'l', 'l', 'o', '\0' }; char message [] = "Hello";
Accepting Strings	scanf and gets can be used to accept strings	char name[20], address[50]; printf("\n Enter your name :"); scanf("%s", name); printf("\n Enter your address :"); gets(address);
Displaying Strings	printf and puts can be used to display strings.	printf("\n The name is %s:", name); printf("\n The address is :"); puts(address);
String functions	All string operations are performed using functions in "string.h". Some of the most commonly used functions are a. strlen – Returns the number of characters in the string (excluding \0) b. strcpy – Copies one string to another c. strcmp – Compares two strings. Returns 0 (equal), +ve (first string > second), -ve (first string <second). It is case sensitive d. strcmpi – Same as strcmp but ignores case e. strcat – Concatenates the second string to the first. Returns the concatenated string. f. strrev – Reverses a string and returns the reversed string. g.strupr – Converts a string to uppercase. h. strlwr – Converts a string to lowercase	#include <string.h> main() { char str1[50], str2[50],str3[100]; printf("\n Give the first string:"); gets(str1); printf("\n Give the second string string:"); gets(str2); if (strlen(str1) == strlen(str2)) { strcpy(str3, strrev(str1)); strcat(str3,strupr(str2)); puts(strupr(str3)); } else puts(strlwr(str2)); }

Write Programs to solve the following problems

Set – A

1. Write a menu driven program to perform the following operations on strings using standard library functions:
a) Length b) Copy c) Concatenation d) Compare
2. Write a C program to accept names from array & search name from array.

Set – B

1. A palindrome is a string that reads the same-forward and reverse.
Example: “madam”, “nayan” are Palindrome strings.
Write a function which accepts a string and returns 1 if the string is a palindrome and 0 otherwise. Use this function in main.
2. Write a C program to Count total number of words in a given string.

Signature of the Instructor

Remark

Date

Assignment:3

Assignment to demonstrate array of strings.

An array of strings is a two dimensional array of characters. It can be treated as a 1-D array such that each array element is a string.

Actions Involving array of strings	Explanation	Example
Declaring String array	Char array[size1][size2];	char cities[4][10]
Initializing String array		char cities[4][10] = { "Pune", "Mumbai", "Delhi", "Chennai"};

Write Programs to solve the following problems

Set – A

1. Write a program to create, read and print an array of strings in C.
2. Write a program that accepts n strings and displays the longest string.

Set – B

1. Write a function, which displays a given number in words.

For Example: 129 One Hundred Twenty Nine

2019 Two Thousand Nineteen

2. Write a program that accepts n words and outputs them in dictionary order.

Signature of the Instructor

Remark

Date

Assignment:4

Assignment to demonstrate use of bitwise operators

Bitwise operators: C provides 6 operators to perform operations on bits. These operators operate on integer and character but not the float and double. Ones complement operator (~) is unary while the others are binary.

Operator	Purpose	Example
~	One's complement	~a : Complements each bit of variable a
>>	Right shift	a=a>>1; Shifts bits of a one position to the right
<<	Left Shift	a=a<<n; Shifts bits of a n positions to the left
&	Bitwise AND	a = b & c; performs bitwise AND on b and c a = a & 0xFF00; Masks the lower order 8 bits of a
	Bitwise OR	a = a b; performs bitwise OR on b and c
^	Bitwise XOR	x = x ^ y; y=x ^ y; x=x ^ y; Swaps x and y by performing bitwise XOR.

Sample code: The following function accepts an integer argument and displays it in binary format. It uses shift operator and AND masking.

```
void displaybits(unsigned int n)
{
    unsigned int mask = 32768; /*set MSB of mask to 1
    */ while (mask>0)
    {
        if((n & mask)==0)
            printf("0");
        else printf("1");

        mask = mask >>1; /* shift mask right */
    }
}
```

Write Programs to solve the following problems

Set – A

1. Write a program to swap two variables without using a temporary variable.(Hint: Use XOR)

2. Write a program which accepts two integers x and y and performs $x \ll y$ and $x \gg y$.
Display the result in binary.

Set – B

1. Write a c program to perform bit-level operations in C programming
(AND, OR, NOT & X-OR)
2. Write a C program to check whether the given number is palindrome or not using bitwise operator.

Signature of the Instructor

Remark

Date

Assignment:5

Assignment to demonstrate structures (using array and functions)

A structure is a composition of variables possibly of different data types, grouped together under a single name. Each variable within the structure is called a 'member'.

Operations performed	Syntax / Description	Example
Declaring a structure	<pre>struct structure-name { type member-1 ; type member-2; . . type member-n ; };</pre>	<pre>struct student { char name[20]; introllno; int marks; };</pre>
Creating structure variables	<pre>struct structurename variable;</pre>	<pre>struct student stud1;</pre>
Accessing structure members	<pre>variable.member</pre>	<pre>stud1.name stud1.rollno stud1.marks</pre>
initializing a structure variable	the initialization values have to be given in {} and in order	<pre>struct student stud1 = {"ABCD",10,95};</pre>
Pointer to a structure	<pre>struct structure-name * pointer-name;</pre>	<pre>struct student *ptr; ptr = &stud1;</pre>
Accessing members using Pointer	<pre>pointer-name -> member-name;</pre>	<pre>ptr->name; ptr->rollno;</pre>
Array of structures	<pre>struct structure-name array-name[size];</pre>	<pre>struct student stud[10];</pre>
passing Structures to Functions	<pre>return-type function-name (struct structure-name variable);</pre>	<pre>void display(struct student s);</pre>
pass an array of structures to a function	<pre>return-type function-name (struct structure-name array[size]);</pre>	<pre>void display(struct student stud[10]);</pre>

Write Programs to solve the following problems

Set – A

1. Write a C program to Accept details for 'n' Student & display information of each Student.
2. Write a C program to accept details of n students (roll no., name, marks of 3 subjects, percentage) and display student data having more than 70%marks.
3. Write a C program to Accept details for 'n' books & write a menu driven program for
 - 1) Display all textbooks
 - 2) Find total cost of all books

Set – B

1. Create a structure student (roll number, name, marks of 3 subjects, percentage). Accept details of n students and write a menu driven program to perform the following operations. Write separate functions for the different option
 - a. Display all student details
 - b. Display all student having percentage > _____
 - c. Display student having maximum percentage
2. Create a structure employee (id, name, salary). Accept details of n employees and write a menu driven program to perform the following operations. Write separate functions for the different options.
 - a. Search byname
 - b. Search by id
 - c. Display all
 - d. Display all employees having salary > _____
 - e. Display employee having maximum salary Instructor should fill in the blanks with appropriate values.

Signature of the Instructor

Remark

Date

Assignment:6

Assignment to demonstrate structures and unions

Operations performed	Syntax	Example
Nested Structure	<pre> struct structure1 { ... struct structure2 { ... } variable; ... }; Method 2 struct structure2 { ... }; struct structure1 { ... struct structure2 variable; ... }; </pre>	<pre> struct student { int rollno; char name[20]; struct date { int dd, mm, yy; } bdate, admdate; }; struct date { int dd, mm, yy; }; struct student { int rollno; char name[20]; struct date bdate, admdate; }; </pre>
Accessing nested structure members	nested structure members can be accessed using the (.) operator repeatedly.	stud1.bdate.dd, stud1.bdate.mm
Self Referential structure	A structure containing a pointer to the same structure	<pre> struct node { int info; struct node *next; }; </pre>
Unions	A union is a variable that contains multiple members of possibly different data types grouped together under a single name. However, only one of the members can be used at a time. They occupy the same memory area.	<pre> unionu { char a; intb; }; </pre>

```
/* Program for demonstrating structure and union */
```

```
struct library_book
{
    int id;
    char title[80],publisher[20] ; int code;
    union u
    {
        int no_of_copies; char month[10];
        int edition;
    } info;
    int cost;
};

void main( )
{
    struct library_book book1;
    printf("\n Enter the details of the book \n");
    printf("\n Enter the id, title and publisher \n");
    scanf("%d%s%s",&book1.id, book1.title, book1.publisher);
    printf("\n Enter the code: 1-Text Book, 2- Magazine, 3-Reference");
    scanf("%d",&book1.code);
    switch(book1.code)
    {
        case 1: printf("Enter the number of copies :");
                scanf("%d",&book1.info.no_of_copies); break;
        case 2: printf("Enter the issue month name :");
                scanf("%s",book1.info.month); break;
        case3 : printf("Entertheeditionnumber:");
                scanf("%d",&book1.info.edition); break;
    }
}
```

```

printf("Enter the cost :");
scanf("%d",&book1.cost);
/* Display details of book */
printf("\n id = %d", book1.id);
printf("\n Title = %s", book1.title);
printf("\n Publisher = %s", book1.publisher);
switch(book1.code)
{
case 1: printf("Copies = %d:", book1.info.no_of_copies); break;
case 2: printf("Issue month name = %s",book1. info.month); break;
case 3: printf("Edition number =%d:",book1.info.edition); break;
}
printf("\n Cost = %d", book1. cost);
}

```

Write Programs to solve the following problems

Set – A

1. Modify the demonstration program above to accept details for n books and write a menu driven program for the following:
 - i) Display all textbooks
 - ii) Search Text Book according to Title
 - iii) Find the total cost of all books (Hint: Use no_of_copies)

2. Modify the sample program 1 to accept details for n books and write a menu driven program for the following:
 - i) Display all magazines
 - ii) Display magazine details for specific month.
 - iii) Find the “costliest” magazine.

Set – B

1. Modify the demonstration program to accept details for n books and write a menu driven program for the following

- i) Display all reference books
- ii) Find the total number of reference books
- iii) Display the edition of a specific reference book.

2. Write a c program to display the student information using structure.

Signature of the Instructor

Remark

Date

Assignment: 7

Assignment to demonstrate command line arguments and preprocessor directives

Preprocessor directives	They begin with a # which must be the first non-space character on the line. They do not end with a semicolon.	
Macro Substitution Directive	# define MACRO value	# define PI 3.142
Argumented Macro	# define MACRO(argument) Value	# define SQR(x) x*x #define LARGER(x, y) ((x)>(y)?(x):(y))
Nested macro	one macro using another	#define CUBE(x) (SQUARE(x)*(x))
File Inclusion directive	#include<filename> #include "filename"	#include <stdio.h>
Conditional Compilation Directive	# if, # else, # elif, # endif #ifdef	#ifdef PI #undef PI #endif
Command Line Arguments	int argc - argument counter char *argv[]-argument vector	void main(int argc, char *argv[]) { printf("There are %d arguments in all", argc); for (i=0; i<argc; i++) printf("Argument %d=%s", i, argv[i]); }
To run a program using command line arguments	Compile the program using cc Execute the program using a.out followed by command line Arguments	Example: a.out ABC 20 Here, ABC and 20 are the two command line arguments which are stored in the form of strings. To use 20 as an integer, use function atoi . Example: int num = atoi (argv [2]);

Write Programs to solve the following problems

Set – A

1. Write a C program to Display contents of file using command line argument.
2. Write a C program to Copy the contents of one file to another Using command line argument.

Set – B

1. Write a program to accept two filenames as command line arguments. Copy the contents of the first file to the second such that the case of all alphabets is reversed.
2. Define a macro EQUALSTR which compares two strings x and y and gives 1 if equal and 0 otherwise. Use this macro to accept two strings from the user and check if they are equal.

Signature of the Instructor

Remark

Date

Assignment: 8

Assignment to demonstrate files handling (text files)

Operations performed	Syntax	Example
Declaring File pointer	FILE * pointer;	FILE *fp;
Opening a File	fopen ("filename", mode); where mode = "r", "w", "a", "r+", "w+", "a+"	fp=fopen("a.txt", "r");
Checking for successful open	if (pointer==NULL)	if(fp==NULL) exit(0);
Checking for end of file	feof	if(feof(fp)) printf("File has ended");
Closing a File	fclose(pointer); fcloseall();	fclose(fp);
Character I/O	fgetc, fscanf, fputc, fprintf	ch=fgetc(fp); fscanf(fp, "%c",&ch); fputc(fp,ch);
String I/O	fgets, fscanf fputs, fprintf	fgets(fp,str,80); fscanf(fp, "%s",str);
Reading and writing formatted data	fscanf fprintf	fscanf(fp, "%d%s",&num,str); fprintf(fp, "%d\t%s\n", num, str);
Random access to Files	ftell, fseek, rewind	fseek(fp,0,SEEK_END); /* end of file*/ long int size = ftell(fp);

Write Programs to solve the following problems

Set – A

1. Write a C program to Display contents of file using file handling.
2. Write a C program read the contents of file and Count no of characters, blanks, tabs and lines from file.
3. Write a C program to Copy the contents of One File to Another File.

Set – B

1. Write a program to accept a filename as command line argument and count the number of words, lines and characters in the file.

2. Write a program to accept details of n students (roll number, name, percentage) and write it to a file named "student.txt". Accept roll number from the user and search the student in the file. Also display the student details having the highest percentage.

Signature of the Instructor

Remark

Date

Assignment: 9

Assignment to demonstrates file handling (binary files and random access to files)

In binary files, information is written in the form of binary. All data is written and read with no interpretation and separation i.e. there are no special characters to mark end of line and end of file. I/O operations on binary files

Reading from a binary file	fread (address, size-of-element, number of elements, pointer);	fread (&num, sizeof(int),1,fp); fread (&emp, sizeof(emp),1,fp); fread(arr,sizeof(int),10,fp);
Writing to a binary File	fwrite (address, size-of-element, number of elements, pointer);	fwrite (&num,sizeof(int),1,fp); fwrite (&emp,sizeof(emp),1,fp);

/* Program to demonstrate binary file */

```
struct employee
{ char name[20]; float sal;
};

main( )
{
    FILE *fp;
    struct employee e; int i;
    if((fp=fopen("employee.in","wb"))==NULL)
        { printf("Error opening file"); exit();
        }

    for(i=0;i<5;i++)
    {
        printf("\n Enter the name and salary");
        scanf("%s%f",e.name,&e.sal); fwrite(&e, sizeof(e),1,fp);
    }
    fclose(fp);
    fp=fopen("employee.in","rb");    /*    reopen file*/
```

```

        if(fp==NULL)
        { fprintf (stderr, "Error opening file); exit();
        }
        for(i=0;i<5;i++)
        {
                fread (&e, sizeof(e),1,fp);
                printf("\n Name = %s Salary = %f", e.name, e.sal);
        } fclose(fp);
}

```

Set – A

1. Create a structure student (roll number, name, percentage) Write a menu driven program to perform the following operations on a binary file-“student.dat”. Write separate functions for the different options.
 1. Add a student(Note: Students should be assigned roll numbers consecutively)
 2. Search Student
 - a) according to name
 - b) according to roll number
 3. Display all students.

Set – B

1. Create a structure student (roll number, name, percentage). Write a menu driven program to perform the following operations on a binary file-“student.dat”. Write separate functions for the different options.
 - a. Add a student(Note: Students will be assigned roll numbers consecutively)
 - b. Delete student
 - i) according to name
 - ii) according to roll number
 - c. Display all students.

Signature of the Instructor

Remark

Date

Assignment: 10

Assignment to demonstrate Graphics Programming.

The graphics.h header file provides access to a simple graphics library that makes it possible to draw lines, rectangles, ovals, arcs, polygons, images, and strings on a graphical window. The second step is initialize the graphics drivers on the computer using initgraph method of graphics.h library.

The first step in any graphics program is to include graphics.h header file. The graphics.h header file provides access to a simple graphics library that makes it possible to draw lines, rectangles, ovals, arcs, polygons, images, and strings on a graphical window.

The second step is initialize the graphics drivers on the computer using initgraph method of graphics.h library.

```
void initgraph(int *graphicsDriver, int *graphicsMode, char  
*driverDirectoryPath);
```

It initializes the graphics system by loading the passed graphics driver then changing the system into graphics mode. It also resets or initializes all graphics settings like color, palette, current position etc, to their default values. Below is the description of input parameters of initgraph function.

- **graphicsDriver** : It is a pointer to an integer specifying the graphics driver to be used. It tells the compiler that what graphics driver to use or to automatically detect the drive. In all our programs we will use DETECT macro of graphics.h library that instruct compiler for auto detection of graphics driver.
- **graphicsMode** : It is a pointer to an integer that specifies the graphics mode to be used. If *gdriver is set to DETECT, then initgraph sets *gmode to the highest resolution available for the detected driver.
- **driverDirectoryPath** : It specifies the directory path where graphics driver files (BGI files) are located. If directory path is not provided, then it will search for driver files in current working directory directory.

1. **getpixel function in c**

getpixel function returns the color of pixel present at location(x, y).

Declaration: `int getpixel(int x, int y);`

2. **getx function in C**

The function getx returns the X coordinate of the current position.

Declaration: `int getx();`

3. **gety function in c**

gety function returns the y coordinate of current position.

Declaration: `int gety();`

4. **line Function** Draws Line From (x1,y1) to (x2,y2) .

Syntax : `line(x1,y1,x2,y2);`

Parameter Explanation

- x1 - X Co-ordinate of First Point
- y1 - Y Co-ordinate of First Point
- x2 - X Co-ordinate of Second Point
- y2 - Y Co-ordinate of Second Point

At the end of our graphics program, we have to unloads the graphics drivers and sets the screen back to text mode by calling closegraph function.

5. **Circle function in c**

Declaration: `void circle(int x, int y, int radius);`

Circle function is used to draw a circle with center (x,y) and third parameter specifies the radius of the circle.

6. **ellipse function in c**

Declarations of ellipse function :-

`void ellipse(int x, int y, int stangle, int endangle, int xradius, int yradius);`

Ellipse is used to draw an ellipse (x,y) are coordinates of center of the ellipse, stangle is the starting angle, end angle is the ending angle, and fifth and sixth parameters specifies the X and Y radius of the ellipse. To draw a complete ellipse strangles and end angle should be 0 and 360 respectively.

7. fillellipse function in C

Declaration of fillellipse function :-

```
void fillellipse(int x, int y, int xradius, int yradius);
```

x and y are coordinates of center of the ellipse, xradius and yradius are x and y radius of ellipse respectively.

Set - A

1. Program to initializes graphics mode
2. Program to Printing Text in Graphics Using Outtextxy Function
3. Program to Draw a line.
4. Program to Draw a Circle.
5. Program to Draw a Rectangle

Set -B

1. Program to draw Moving On circles (concentric circles)
2. Program to draw Basic Shapes and Colors
3. Program to draw Moving Car.

Signature of the Instructor
Date

Remark

C Programming – Case Study

(Handling various string operations without built in functions)

Design a program in C to cover the following options:

1. Find the length of the string using pointer.
2. Find whether the given string is present in large string or not
3. Compare two strings without using library functions
4. Count total number of capital and small letters from accepted line using point
5. Concatenate two strings without using library function.

Signature of the Instructor

Remark

Date

Anekant Education Society's
Tuljaram Chaturchand College of
Arts, Science & Commerce Baramati(Autonomous)

DEPARTMENT OF COMPUTER SCIENCE

Computer Practical Journal

Paper IV: CSCO – 1204 Lab Course on DBMS-II

- CERTIFICATE -

University Seat No.

Date :

This is to certify that Mr./ Miss. / Smt. _____
_____ has satisfactorily completed the course in practical as
prescribed by the University of Pune for the **F.Y.B.Sc.(Computer Science-Lab
Course II** in the Year 2019 – 2020

In-Charge
(Practical)

Internal Examiner

External Examiner

Head
Computer Science

Index

Sr.No.	Title of Experiment/ Practical	Date	Sign
1	Simple Queries		
2	Nested Queries, using aggregate functions		
3	Queries using Views		
4	Stored Function		
5	Cursors		
6	Exception Handling		
7	Triggers		
8	Case Study		

Assignment: 01

Set A

Person –Area Database

Consider the relation Person (P_number, P_name, Birthdate, Income), Area(Aname, Area_type). An area can have one or more persons living in it, but a person belongs to exactly one area. The attribute 'Area_type' can have values either 'Urban' or 'Rural'. Create the relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF).

Insert sufficient number of appropriate records.

Solve the Queries

1. List the names of all people living in 'Urban' area.
2. List the details of all people whose names start with the alphabet 'A'.
3. List the names of all people whose birthday falls in the month of March.
4. Count the people who are born on 6/7/1997.
5. Count the people whose income is below 5000.
6. List the names of all people whose income is between 5000 and 6000.
7. List the names of people with average income.
8. List the details of the people, sorted by person name.
9. Transfer all people living in 'Pune' to 'Mumbai'.
10. Delete information of people staying in 'urban' area.

Set B

Movie Database

Movies(M_name, release_year, budget)

Actor(A_name, role, charges, A_address)

Producer(producer_id, name, P_address)

Each actor has acted in one or more movies. Each producer has produced many movies and each movie can be produced by more than one producers. Each movie has one or more actors acting in it, in different roles. Create the relations accordingly, so that the relationship is handled properly and the relations are in normalized form(3NF). Insert sufficient number of appropriate records.

Solve the Queries:

1. List the names of actors who have acted in at least one movie, in which ' _____ ' has acted.
2. List the names of the movies with the highest budget.
3. List the names of actors who have acted in the maximum number of movies.
4. List the names of movies, produced by more than one producer.
5. List the names of actors who are given with the maximum charges for their movie.
6. List the names of producers who produce the same movie as ' _____ '.
7. List the names of actors who do not live in _____ or _____ city.

Signature of the Instructor

Remark

Date

Assignment: 02

Set A

1. Bank database

Consider the following database maintained by a Bank. The Bank maintains information about its branches, customers and their loan applications. Following are the tables:

BRANCH (BID INTEGER, BRNAME VARCHAR (30), BRCITY CHAR (10)),

CUSTOMER (CNO INTEGER, CNAME VARCHAR (20), CADDR VARCHAR (35), CITY VARCHAR(20)),

LOAN_APPLICATION (LNO INTEGER, L_AMT_REQUIRED float, L_AMT_APPROVED float, L_DATE DATE)

The relationship is as follows:

BRANCH, CUSTOMER, LOAN_APPLICATION are related with ternary relationship.

TERNARY (BID INTEGER, CNO INTEGER, LNO INTEGER).

Solve the Queries:

1. List the names of the customers who have received loan less than their requirement.
2. Find the maximum loan amount approved.
3. Find out the total loan amount sanctioned by “Deccan” branch.
4. Count the number of loan applications received by “M.G.ROAD” branch.
5. List the names of the customer along with the branch names who have applied for loan in the month of November.

2. Student- Teacher database

Consider the following database maintained by a school. The school maintains information about students and the teachers. It also gives information of the subject taught by the teacher. Following are the tables:

STUDENT (SNO INTEGER, S_NAME VARCHAR(30), S_CLASS VARCHAR(10), S_ADDR VARCHAR(50))

TEACHER (TNO INTEGER, T_NAME VARCHAR (20), QUALIFICATION VARCHAR (15), EXPERIENCE INTEGER)

The relationship is as follows:

STUDENT-TEACHER: M-M with descriptive attribute SUBJECT.

Solve the queries

1. Find the minimum experienced teacher.
2. Find the number of teachers having qualification “Ph. D.”.
3. List the names of the students to whom “kavita” is teaching along with the subjects he is teaching to them.
4. List the names of the teachers who are teaching to a student named “Asmita”.
5. List the names of all teachers along with the total number of students they are teaching.

Set B

1. Project-Employee database

Consider the database maintained by a company which stores the details of the projects assigned to the employees.

Following are the tables:

Project (pno integer, p_name varchar(30), ptype varchar(20), duration integer)

Employee (eno integer, e_name varchar (20), qualification varchar (15), joindate date)

The relationship is as follows:

Project - Employee: M-M relationship , with descriptive attributes as start_date (date), no_of_hours_worked (integer).

Solve the queries:

1. Find the names of the employees starting with 'A'.
2. Find the details of employees working on the project “Operating system”.
3. Find the employee numbers of the employees, who do not work on project “Robotics”.
4. List the names of the first three employees in alphabetical order.
5. Find the names of the employees who have worked for more than three hours.

2. Business -Trip database

Consider the business trip database that keeps track of the business trips of salesman in office. Following are the tables:

salesman (sno integer, s_name char (30), start_year year, deptno varchar (10))

trip (tno integer, from_city char (20), to_city char (20), departure_date date, return date)

dept (deptno varchar (10), dept_name char(20))

expense (eid integer, amount float)

The relationship is as follows

dept-salesman 1 to m

salesman - trip 1 to m

Trip - Expense 1 to 1

Solve the queries:

1. Give the details for trips that exceed rs. 10,000 in expenses.
2. List the salesman numbers and names of the salesmen who made trips to daund.
3. Find the total expenses incurred by the salesman "mr. patil".

Signature of the Instructor

Remark

Date

Assignment 03 : Views

Views can be thought of as stored queries, which allow us to create a database object that functions very similarly to a table, but whose contents dynamically reflect the selected rows. Views are very flexible; you may use a view to address common simple queries to a single table, as well as for complicated ones which may span across several tables.

Creating a View: CREATE [OR REPLACE] [TEMP | TEMPORARY] VIEW *name* [(*column_name* [...])] AS *query*

CREATE VIEW defines a view of a query. The view has no physical existence, but is dynamically created whenever it is referenced in a query.

CREATE OR REPLACE VIEW replaces an existing view of same name. The new query must use the same column names in the same order and with the same data types, but it may add additional columns to the end of the list.

Schema name (CREATE VIEW myschema.myview) must be specified to create a view in schema other than the current schema. The TEMPORARY or TEMP parameter is specified to create Temporary views, however If any of the tables referenced by the view are temporary, the view is created as a temporary view (whether TEMPORARY is specified or not). Temporary views exist in a special schema, so a schema name cannot be given when creating a temporary view. View names should be distinct.

Name: Name (optionally schema-qualified) of a view to be created.

column_name: An optional list of names to be used for columns of the view. If not given, the column names are deduced from the query.

Query: A SELECT command which will provide the columns and rows of the view.

Example to create a view consisting of all comedy films

```
CREATE VIEW comedies AS SELECT * FROM films WHERE type = 'Comedy';
```

ALTER VIEW statement is used to change the definition of a view.

```
ALTER VIEW name ALTER [COLUMN] column SET DEFAULT expression
```

ALTER VIEW *name* ALTER [COLUMN] *column* DROP DEFAULT

DROP VIEW statement is used to remove a view

DROP VIEW [IF EXISTS] *name* [...] [CASCADE | RESTRICT]

Set A

1.Using Project-Employee database

1. Create a view over the employee table which contains only employee name and his qualification and it should be sorted on qualification.
2. Create a view containing the project name, project type and start date of the project and should be sorted by start date of the project.
3. Write the following Queries, on the above created views :
 - a. List different qualifications of employees.
 - b. List the name and type of projects started on 1st April 2014.
 - c. List the names of employees who are qualified as Engineers.

2.Using the Business trip database

1. Create a view to list the details of all salesman from 'Western' department.
2. Create a view to list all salesman's name his trip details and his expenses for the trip.
3. Write the following Queries, on the above created views:
 - a. List the details of salesmen from 'Western' department, whose start year is 2005.
 - b. List the names of salesmen from 'Western' department, for whom their total trip expense is > Rs 100000.
 - c. List the names of salesmen who have gone on a trip to "Mumbai" city.

Set B

1.Using the Warehouse database

1. Create a view containing details of all the stores of a Warehouse named 'Spares'.
2. Create a view to list the details of all customers who have placed orders on the date '03- 10-2013'

3. Write the following Queries, on the above created views :
 - a. List the names of stores of 'Spares' warehouse, located at "Pune".
 - b. List the names of customers from "Pune" city who have placed orders on 03-1-2013 .
 - c. List the orders placed by "Mr. Joshi".

2.Using the Student- Teacher database

1. Create a view containing details of all the teachers teaching the subject 'Mathematics'.
2. Create a view to list the details of all the students who are taught by a teacher having experience of more than 3 years.
3. Write the following Queries, on the above created views :
 - a. List the name of the most experienced teacher for "Mathematics".
 - b. List the names of students of 'S.Y.B.Sc' class, who are taught by a teacher having more than 3 years experience.

Signature of the Instructor

Remark

Date

Assignment: 04

SET A

1.Using Bank Database

1. Write a function that returns the total number of customers of a particular branch.
(Accept branch name as input parameter.)
2. Write a function to find the maximum loan amount approved.

2.Using Project-Employee database

1. Write a function to accept project name as input and returns the number of employees working on the project.
2. Write a function to find the number of employees whose date of joining is before '03/10/2010'

Set B

1. Using Business trip database

1. Write a PL/pgsql function to find a business trip having maximum expenses.
2. Write a PL/pgsql function to count the total number of business trips from 'Pune' to 'Mumbai'.

2. Using Railway Reservation Database

Consider a railway reservation Database of passengers. Passengers reserve berths of a bogie of trains. The bogie capacity of all the bogies of a train is same.

1. TRAIN (TRAIN_NO INT, TRAIN_NAME VARCHAR(20), DEPART_TIME TIME , ARRIVAL_TIME TIME, SOURCE_STN VARCHAR(20) , DEST_STN VARCHAR (20), NO_OF_RES_BOGIES INT , BOGIE_CAPACITY INT)
2. PASSENGER (PASSENGER_ID INT, PASSENGER_NAME VARCHAR(20), ADDRESS VARCHAR(30), AGE INT , GENDER CHAR)

Relationship is as follows:

TRAIN _PASSENGER : M-M with descriptive attributes as follows :

TICKET (TRAIN_NO INT , PASSENGER_ID INT, TICKET_NO INT COMPOSITE KEY,BOGIE_NO INT, NO_OF_BERTHS INT , DATE DATE , TICKET_AMT DECIMAL(7,2),STATUS CHAR)

The status of a particular berth can be 'W' (waiting) or 'C' (confirmed).

a) Write a PL/pgsql function to calculate the ticket amount paid by all the passengers on 13/5/2009 for all the trains.

b) Write a PL/pgsql function to update the status of the ticket from "waiting" to "confirm" for passenger named "Mr. Jadhav"

Drivers are allotted to the buses shiftwise.

Signature of the Instructor

Remark

Date

Assignment: 05

PL/SQL Cursors provide a way to select multiple rows of data from the database and then to process each row individually. Using a cursor, we can traverse up and down a result set and retrieve only those rows which are explicitly requested. Cursors basically help an application to efficiently use a static result set.

Declaring Cursor Variables

All access to cursors in PL/pgSQL goes through cursor variables, which are always of the special data type refcursor. We can declare a cursor variable either as a bound cursor variable or an unbound cursor variable.

- a. To create an unbound cursor variable, just declare it as a variable of type refcursor.
- b. To create a bound cursor variable, use the following cursor declaration syntax

```
name CURSOR [ ( arguments ) ] FOR query;
```

where arguments, if specified, is a comma-separated list of pairs 'name datatype' that define names to be replaced by parameter values in the given query. The actual values to substitute for these names will be specified later, when the cursor is opened (parameterized cursors).

Opening Cursors

Before a cursor can be used to retrieve rows, it must be opened. PL/pgSQL has three forms of the OPEN statement, two of which use unbound cursor variables while the third uses a bound cursor variable.

Syntax for OPEN FOR query

```
OPEN unbound_cursorvar FOR query;
```

Syntax for OPEN FOR EXECUTE

```
OPEN unbound_cursorvar FOR EXECUTE query_string USING  
expression [, ... ];
```

Syntax for Opening a Bound Cursor

```
OPEN bound_cursorvar [ ( argument_values ) ];
```

Using Cursors

Once a cursor has been opened, it can be manipulated with the statements described below.

FETCH

Syntax :

```
FETCH [ direction { FROM | IN } ] cursor INTO target;
```

The direction clause can be any of the following variants:

NEXT, PRIOR, FIRST, LAST, ABSOLUTE count, RELATIVE count, FORWARD, or BACKWARD. Default is NEXT.

MOVE: MOVE repositions a cursor without retrieving any data.

Syntax :

```
MOVE [ direction { FROM | IN } ] cursor;
```

The direction clause can be any of the variants NEXT, PRIOR, FIRST, LAST, ABSOLUTE

count, RELATIVE count, ALL, FORWARD [count | ALL], or BACKWARD [count | ALL].

CLOSE: CLOSE closes the portal underlying an open cursor. This can be used to release

resources earlier than end of transaction, or to free up the cursor variable to be opened again.

Syntax :

```
CLOSE cursor;
```


Practice Examples:

Example 1:

Consider a relation, Employee (eno, ename, deptno,salary). We write a function, to Define a cursor to print the details of the employee along with commission earned for each employee. Commission is 20% of salary for employees of dept no = 5; its 50% of salary for employees of dept no = 8; its 30% of salary for employees of dept no = 10.

Create function cursor_demo() returns integer as ‘

Declare

Emp-rec Employee%rowtype

*C-emp cursor for Select * from employee;*

Comm Number (6,2);

Begin

Loop

Fetch C-emp into emp-rec;

If emp-rec.deptno = 5 then

*Comm:= emp-rec.salary * 0.2;*

Else if emp-rec.deptno = 8 then

*Comm := emp-rec.salary * 0.5;*

Else

If emp-rec.deptno = 10 then

*Comm := emp-rec.salary * 0.3;*

End if;

End if;

Endif;

Raise notice “emp-rec.ename||emp-rec.deptno||emp-rec.salary||comm. “;

Exit when not found;

End loop;

Close C-emp;

End; ' language 'plpgsql';

Example 2:

Consider the following relations,

Dept(dno, dname) Employee(eno, dno, ename, salary) and the relation between Dept and Employee is one to many(1:M). Now we write a PL/pgSQL function to print the list of employees, department wise. Here we use 2 cursors, the 1st Cursor retrieves the department Information for each department and the 2nd cursor, to which dept number is sent as a parameter retrieves the employees for that department.

Create function parameterizedcursor_demo() returns integer as '

Declare

*d-Info cursor for Select * from dept;*

E-Info cursor (dnum dept.dno%type) for

*Select * From Employee Where dno = dnum;*

X number := 0;

Begin

For drec In d-Info

Loop

Raise notice " drec.dno || drec.dname";

For erec In E-Info (drec.dno)

Loop

Raise notice "erec.eno || erec.ename || erec.salary";

X := X + 1;

```

    End loop;

    Raise notice "Total employees for: ||drec.dname|| 'is %' || X";

    End loop;

    Return (X);

End; ' language 'plpgsql';

```

Set A

1. Using railway reservation database

- a) Write a stored function using cursors to find the confirmed bookings of all the trains on 18-05-2009.
- b) Write a stored function using cursors to find the total number of berths not reserved for all the trains on 18-05-2009.

2. Using bus driver database

- a) Write a stored function using cursors to display the details of a driver, (Accept driver name as input parameter).
- b) Write a stored function using cursors to display the details of the buses that run on routes 1,2. (Use two different cursors for route_no = 1 and route_no = 2).

Set B

1. Using Student –Marks database

Student (rollno integer,name varchar(30),address varchar(50),class varchar(10))

Subject(Scode varchar(10),subject name varchar(20))

Student and Subject are related with M-M relationship with attributes marks scored.

Create a RDB in 3NF for the above and solve the following.

- a) Write a stored function using cursors, to accept a address from the user and display the name, subject and the marks of the students staying at that address.

2. Using the Warehouse database

a) Write a stored function using cursors to accept a city from the user and to list all warehouses in the city.

b) Write a stored function using cursors to find the list of items whose cost is between Rs.5000 to 10000

Signature of the Instructor

Remark

Date

Assignment: 06

SET A

1. Using Bank Database(Refer Asss2)

1. Write a stored function to print the total number of customers of a particular branch. (Accept branch name as input parameter.) In case the branch name is invalid, raise an exception for the same.
2. Write a stored function to increase the loan approved amount for all loans by 20%. In case the initial loan approved amount was less than Rs 10000, then print a notice to the user, before updating the amount .

2. Using Project-Employee database(Refer Asss2)

1. Write a stored function to accept project name as input and print the names of employees working on the project. Also print the total number of employees working on that project. Raise an exception for an invalid project name.
2. Write a stored function to decrease the Hours_worked by 2 hours, for all projects in which employees from department no 2 is working. Raise an exception , in case the hours_worked becomes = 0 , after updation.

SET B

1. Using Bus transport Database

- a) Write a stored function to print the names of drivers working on both shifts on '20/04/2014'.
- b) Write a stored function to accept the bus_no and date and print its allotted drivers. Raise an exception in case of invalid bus number.

2. Using Railway Reservation Database(Refer Ass4)

- a) Write a stored function to accept the train_no and date and print its allotted Source_Stn. Raisean exception in case of invalid train number.
- b) Write a stored function to accept the train_no and date and print its allotted Dest_Stn. Raisean exception in case of invalid ticket number.

Signature of the Instructor

Remark

Date

Assignment: 07

A trigger defines a function which occurs before or after , an action on a table. A trigger is implemented through PL/pgSQL, C or any other functional language that PostgreSQL can use to define a function..

A trigger is a PL/pgSQL block that is associated with a table, stored in a database and executed in response to a specific data manipulation event. Triggers can be executed or fired in response to the following events

- a. A row is inserted into table
- b. A row in a table is updated.
- c. A row in a table is deleted

Syntax for defining a database trigger

Create Trigger trigger–name

```
{ Before | After} {event [ or event ...]} ON table–name for each { Row | statement}  
execute procedure functionname ( arguments) ;
```

A trigger procedure is created with the CREATE FUNCTION command, declaring it as a function with no arguments and a return type of trigger.

Special variables created automatically, on call to a trigger function, are as follows :

Variable Name	Purpose and contents
NEW	Holds the new database row for INSERT/UPDATE operations in row-level triggers
OLD	Holds the old database row for UPDATE/DELETE operations in row-level triggers.
TG_NAME	Contains the name of the trigger actually fired.
TG_WHEN	Specifies the trigger timing. Contains a string of BEFORE or AFTER

TG_LEVEL	Specifies the trigger type. Contains a string of either ROW or STATEMENT
TG_OP	Specifies the trigger operation. Contains a string of INSERT, UPDATE, or DELETE.
TG_RELID	Contains the object ID of the table that caused trigger invocation.
TG_TABLE_NAME	Contains the name of the table that caused the trigger invocation.
TG_TABLE_SCHEMA	Contains the name of the schema of the table that caused trigger invocation
TG_NARGS	Number of arguments given to the trigger procedure
TG_ARGV[]	Contains the arguments for the trigger statement.

Practice Examples :

Example 1

This trigger ensures that any time a row is inserted or updated in the table, the current user name and time are stamped into the row. And it checks that an employee's name is given and that the salary is a positive value.

```
CREATE TABLE employee (
  ename text,
  esalary integer,
  last_date timestamp,
  last_user text          );
CREATE FUNCTION emp_timestamp() RETURNS trigger AS '
```

```

BEGIN
-- Check that empname and salary are given
IF NEW.ename IS NULL THEN
RAISE EXCEPTION 'empname cannot be null';
END IF;

IF NEW.esalary IS NULL THEN
RAISE EXCEPTION '% cannot have null salary', NEW.ename;
END IF;

IF NEW.esalary < 0 THEN
RAISE EXCEPTION '% cannot have a negative salary', NEW.ename;
END IF;

-- Remember who changed the payroll when
NEW.last_date := current_timestamp;
NEW.last_user := current_user;
RETURN NEW;
END;
' LANGUAGE 'plpgsql';

```

```

CREATE TRIGGER emp_stamp BEFORE INSERT OR UPDATE ON employee
FOR EACH ROW EXECUTE PROCEDURE emp_stamp();

```

SET A

1. Using Railway Reservation Database

- a) create a trigger to validate train arrival time must be less than train departure time.
- b) Write a trigger which will be activated before changing the status field in the Ticket table and print a message to the user.

2. Using Bus Transportation database

- a) Define a trigger after insert or update the record of driver if the age is between 18 and 50 give the message "valid entry" otherwise give appropriate message.

- b) Define a trigger after delete the record of bus having capacity < 10. Display the message accordingly

SET B :

1. Using Bank Database

- a) create a trigger which contains the details of all customers who had applied for loan more than 200000.
- b) define a trigger delete the record of branch whose city name is 'pune' and display the message.

Signature of the Instructor

Remark

Date

Assignment: 08

A music company wants to go in for automation of their requirements. They want to develop a database for maintaining the information of their music albums, singers, musicians, instruments. The following facts are relevant:

- a. Each album is produced by many musicians, a musician can produce many albums
- b. A singer can sing for many albums, but an album consists of songs of only one singer.
- c. A musician can play many instruments, an instrument can be played by many musicians.

The following constraints are to be placed on the relations

- a. each musician is paid a minimum of 50000 Rs. for each album
- b. all singers are from either pune, Mumbai or Chennai
- c. each instrument cost is maximum 10000

Design the relational database for the above, so that the following queries can be answered:

1. Write a function to find out names of musicians who palsy at least one instrument same as the one "Joshi" plays.
2. Write a trigger after insert or update the record the if the albums released in 1998.
3. Create a view which contains details of the names of albums that have more than two Instruments being played in it.

Signature of the Instructor

Remark

Date